

## Activity 13

---

### Comp 11 - Summer Session — Complexity Analysis

---

With a partner(or two), discuss the following code sample and answer the questions below. The instructor and teaching assistants will let you discuss and then be around to answer questions. <sup>1</sup>

#### 13.1 Description

For this activity, take a look at the following code and figure out what the Big O(worst-case analysis) complexity is of each code sample.

#### 13.2 Questions

1. Why do we generally care most about worst-case analysis(Big-O notation) when discussing algorithms? (That is not to say that other analysis techniques have value however!)
2. What is the complexity of bubble sort?
3. What is the complexity of binary search (assuming the data we are searching is sorted?)

```
4 // O(n) for lines 2-4: -----  
2 for(int n = 0; n < 100; n++){  
3     sum = sum + n;  
4 }
```

Listing 13.1: Complexity 1

---

<sup>1</sup>Activities do not need to be returned to instructors, they are for your benefit.

```

1 // O(n) for line 2: -----
2 int x = 5 * 2 + 4 / 7 * 9 + 400;

```

Listing 13.2: Complexity 2

```

6 // O(n) for lines 2-8: -----
2 for(int n = 0; n < 100; n++){
3     sum = sum + n;
4 }
5
6 for(int m = 0; m < 350; m++){
7     sum = sum + m;
8 }

```

Listing 13.3: Complexity 3

```

1 // O(n) for lines 8-11: -----
2 void foo(){
3     for(int n = 0; n < 100; n++){
4         sum = sum + n;
5     }
6 }
7
8 // Analyze lines 8-10
9 for(int m = 0; m < 350; m++){
10     foo();
11 }

```

Listing 13.4: Complexity 4

```

8 // O(n) for lines 2-9: -----
2 int iterations = 255;
3 for(int n = 0; n < iterations; n++){
4     for(int m = 0; m < iterations; m++){
5         for(int o = 0; o < iterations; o++){
6             std::cout << "Hello\n";
7         }
8     }
9 }

```

Listing 13.5: Complexity 5